

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-091456

(43)Date of publication of application : 10.04.1998

(51)Int.Cl.

G06F 9/45

(21)Application number : 09-140445

(71)Applicant : FUJITSU LTD

(22)Date of filing : 29.05.1997

(72)Inventor : GOTO RIE

(30)Priority

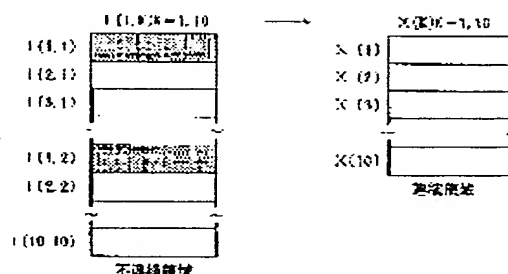
Priority number : 08138792 Priority date : 31.05.1996 Priority country : JP

(54) METHOD AND SYSTEM FOR PROCESSING INPUT/OUTPUT REPETITION OF PROGRAM, AND STORAGE MEDIUM FOR STORING INPUT/OUTPUT REPETITION PROCESSING PROGRAM OF PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To speed up CPU performance and to guarantee the unification of a program to a user by replacing an uncontinuous area with a continuous area inside a library even when the area of the library to be accessed is uncontinuous and defining the calling time of a processing module in an input/output repetition processing as one.

SOLUTION: A value copied from the uncontinuous area to the continuous area becomes  $I(1, 1) \sim I(1, 10)$  based on a prescribed parameter and the values are copied to the  $X(1) \sim X(10)$  of the continuous area. Thus, a repetition processing module issues an execution instruction to a simple item processing module. The simple item processing module performs access to the input/output object item of the continuous area in the library. That is, the repetition processing module generates the continuous area and it is adequate that the simple item processing module is called only once. Thus, a user can select the re-editing processing of the value to be stored in the uncontinuous area.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-91456

(43) 公開日 平成10年(1998) 4月10日

(51) Int.Cl.<sup>6</sup>

識別記号

F I

G 0 6 F 9/45

G 0 6 F 9/44

3 2 2 G

審査請求 未請求 請求項の数16 O L (全 12 頁)

(21) 出願番号 特願平9-140445

(71) 出願人 000005223

(22) 出願日 平成9年(1997) 5月29日

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(31) 優先権主張番号 特願平8-138792

(72) 発明者 後藤 理恵

(32) 優先日 平8(1996) 5月31日

静岡県静岡市伝馬町16番地の3 株式会社

(33) 優先権主張国 日本 (J P)

富士通静岡エンジニアリング内

(74) 代理人 弁理士 伊東 忠彦

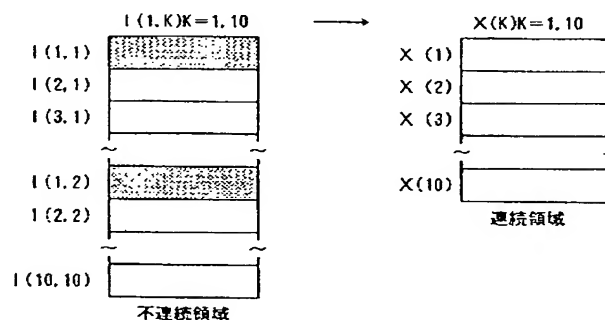
(54) 【発明の名称】 プログラムの入出力反復処理方法及びシステム及びプログラムの入出力反復処理プログラムを格納した記憶媒体

(57) 【要約】 (修正有)

【課題】 プログラムに反復記述された場合でも、不連続領域に格納されているデータを再編することを可能とし、1回の処理モジュールの呼出しで実行可能なプログラムとする。

【解決手段】 プログラム中に繰り返し文が存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータに指定し、入出力対象項目要素が入出力ライブラリ内の不連続領域に存在する場合に、オブジェクトインターフェースに指定されたパラメータに基づいて、入出力ライブラリ内部で不連続領域の値を連続領域に置き換え、連続領域をアクセスするオブジェクトプログラムとする。

本発明の第1の実施例のFORTRANプログラムの繰り返し処理を説明するための図



## 【特許請求の範囲】

【請求項1】 FORTRAN言語、BASIC言語または、C言語を含むプログラム中に入出力反復の入出力が記載されており、入出力ライブラリ内の不連続領域をアクセスするプログラムの入出力反復処理方法において、

前記プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定し、

入出力対象項目が前記入出力ライブラリ内の前記不連続領域に存在する場合に、前記オブジェクトインターフェースに指定されたパラメータに基づいて、前記入出力ライブラリ内部で前記不連続領域の値を連続領域に置き換え、

前記連続領域をアクセスするようオブジェクトプログラムを制御することを特徴とするプログラムの入出力反復処理方法。

【請求項2】 前記オブジェクトインターフェース中のパラメータの1つの要素の長さと、増加アドレス値と比較し、

前記1つの要素の長さと、前記増加アドレス値が一致しない場合には、前記入出力ライブラリ内の前記不連続領域の値を格納するための連続領域を確保し、

確保された前記連続領域に、前記オブジェクトインターフェースに指定された前記パラメータに基づいて前記不連続領域の値を複写する請求項1記載のプログラムの入出力反復処理方法。

【請求項3】 前記不連続領域の値を前記連続領域に格納する際に、

指定された前記不連続領域の値の有効値のみを選択して、前記連続領域に格納する値とする請求項1及び2記載のプログラムの入出力反復処理方法。

【請求項4】 前記不連続領域の値を前記連続領域に格納する際に、

指定された前記連続領域へ置換する前記不連続領域の有効値から次の有効値までの配列要素数及び置換個数に基づいて、前記不連続領域から前記連続領域への置換を行う請求項1及び2記載のプログラムの入出力反復処理方法。

【請求項5】 FORTRAN言語、BASIC言語または、C言語を含むプログラム中に入出力反復の入出力が記載されており、入出力ライブラリ内の不連続領域をアクセスするプログラムの入出力反復処理方法において、

前記プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定し、

入出力対象項目が前記入出力ライブラリ内の前記不連続

領域に存在する場合に、前記オブジェクトインターフェースに指定されたパラメータに基づいて、繰り返し対象項目をアクセスするモジュールに、前記入出力ライブラリ内部で有効となる項目のパラメータを渡すことを特徴とするプログラムの入出力反復処理方法。

【請求項6】 前記繰り返し対象項目をアクセスするモジュールに、

指定された前記不連続領域の値の有効値のみを選択して、前記連続領域に格納する値を渡す請求項5記載のプログラムの入出力反復処理方法。

【請求項7】 前記繰り返し対象項目をアクセスするモジュールに、

指定された前記連続領域へ置換する前記不連続領域の有効値から次の有効値までの配列要素数及び置換個数を渡す請求項5記載のプログラムの入出力反復処理方法。

【請求項8】 入出力ライブラリ内の不連続領域をアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、前記プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定するパラメータ指定手段と、

前記繰り返し項目が前記入出力ライブラリ内の前記不連続領域に存在する場合に、前記パラメータ指定手段により指定されたパラメータに基づいて、前記入出力ライブラリ内部で前記不連続領域の値を連続領域に置き換える連続領域置換手段と、

前記連続領域をアクセスするようオブジェクトプログラムを制御するオブジェクトプログラム制御手段とを有することを特徴とするプログラムの入出力反復処理システム。

【請求項9】 前記連続領域置換手段は、前記オブジェクトインターフェース中のパラメータの1つの要素の長さと、増加アドレス値を比較する比較手段と、

前記1つの要素の長さと、前記増加アドレス値が一致しない場合には、前記入出力ライブラリ内の前記不連続領域の値を格納するための連続領域を確保する連続領域確保手段と、

確保された前記連続領域に、前記オブジェクトインターフェースに指定された前記パラメータに基づいて前記不連続領域の値を複写する複写手段とを含む請求項8記載のプログラムの入出力反復処理システム。

【請求項10】 前記連続領域置換手段は、前記不連続領域の値を前記連続領域に格納する際に、前

記不連続領域の有効値を指定する有効値指定手段と、前記有効値指定手段により指定された前記不連続領域の値の有効値のみを選択して、前記連続領域に格納する値とする第1の置換値決定手段とを含む請求項8及び9記載のプログラムの入出力反復処理システム。

【請求項11】 前記連続領域置換手段は、前記不連続領域の値を前記連続領域に格納する際に、前記連続領域へ置換する前記不連続領域の有効値から次の有効値までの配列要素数及び置換個数を指定する置換値指定手段と、前記置換値指定手段により指定された値に基づいて、前記不連続領域から前記連続領域への置換を行う値を決定する第2の置換値決定手段とを含む請求項8及び9記載のプログラムの入出力反復処理システム。

【請求項12】 入出力ライブラリ内の不連続領域にアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、

前記プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定するパラメータ指定手段と、

前記繰り返し項目が前記入出力ライブラリ内の前記不連続領域に存在する場合に、前記パラメータ指定手段により指定されたパラメータに基づいて、前記入出力ライブラリ内部で前記不連続領域のアクセス対象となる項目をアクセスするためのパラメータを生成するアクセス対象パラメータ生成手段と、

前記アクセス対象パラメータ生成手段により生成されたパラメータを、繰り返し処理を行うオブジェクトに転送するパラメータ転送手段とを有することを特徴とするプログラムの入出力反復処理システム。

【請求項13】 前記アクセス対象パラメータ生成手段は、

前記不連続領域内のアクセス対象となる有効値を指定する有効値指定手段を含む請求項12記載のプログラムの入出力反復処理システム。

【請求項14】 前記アクセス対象パラメータ生成手段は、

前記不連続領域内のアクセス対象となるある有効値から次の有効値までの配列要素数及び置換個数または、バイト数を指定する置換値指定手段を含む請求項12記載のプログラム入出力反復システム。

【請求項15】 入出力ライブラリ内の不連続領域にアクセスするFORTRAN言語、BASIC言語また

は、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、

前記プログラム中の入出力文において、繰り返し項目の並びを解析する解析プロセスと、

繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定させるパラメータ指定プロセスと、

前記繰り返し項目が前記入出力ライブラリ内の前記不連続領域に存在する場合に、前記パラメータ指定プロセスにより指定されたパラメータに基づいて、前記入出力ライブラリ内部で前記不連続領域の値を連続領域に置き換える連続領域置換プロセスと、

前記連続領域にアクセスするようオブジェクトプログラムを制御するオブジェクトプログラム制御プロセスとを有することを特徴とするプログラムの入出力反復処理プログラムを格納した記憶媒体。

【請求項16】 入出力ライブラリ内の不連続領域にアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、

前記プログラム中の入出力文において、繰り返し項目の並びを解析する解析プロセスと、

繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定させるパラメータ指定プロセスと、

前記繰り返し項目が前記入出力ライブラリ内の前記不連続領域に存在する場合に、前記パラメータ指定プロセスにより指定されたパラメータに基づいて、前記入出力ライブラリ内部で前記不連続領域のアクセス対象となる項目をアクセスするためのパラメータを生成するアクセス対象パラメータ生成プロセスと、

前記アクセス対象パラメータ生成プロセスにより生成されたパラメータを繰り返し処理を行うオブジェクトに転送するパラメータ転送プロセスとを有することを特徴とするプログラムの入出力反復処理プログラムを格納した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プログラムの入出力反復処理方法及びシステム及びプログラムの入出力反

10

20

30

40

50

5

復処理プログラムを格納した記憶媒体に係り、特に、プログラムに記述されている反復処理をオブジェクトにおいて、n回繰り返すようなプログラムの入出力反復処理方法及びシステム及びプログラムの入出力反復処理プログラムを格納した記憶媒体に関する。

【0002】例えば、FORTRANで記述されたプログラムに記述される入出力DO形反復である入出力項目が記述された繰り返し処理の効率化を図るためのプログラムの入出力反復処理方法及びシステム及びプログラムの入出力反復処理プログラムを格納した記憶媒体に関する。

【0003】

【従来の技術】以下に従来の技術をFORTRANを例に説明する。FORTRANプログラムにおける入出力DO形反復の形式は、以下のような形式がある。

① 入出力DO形反復は、入出力DO形項目及び入出力DO制御である。

② 入出力DO項目並びは、変数または、式である。

③ 入出力DO制御は、DO形変数＝スカラ数値式、スカラ数値[、スカラ数値式]である。

【0004】例えば、以下の例1のFORTRANプログラムか、

(例1)

```
REAL*4 A(100,100)
L=1
WRITE(10) (A(L,I),I=1,100)
...
END
```

であるとき、配列の並びとアクセス及びモジュールの呼び出しの関係では、入出力並びに指定された配列Aは、各要素が図7に示すようにライブラリ内に配置されており、“WRITE”文実行時にオブジェクトプログラムは、aの部分の配列要素をアクセスする。このaの部分の配列要素のアクセス毎に、図8に示す単純項目処理モジュールAが呼び出される。

【0005】つまり、図7において、上記の“WRITE(10) (A(L,I),I=1,100)”文がコンパイルされ、オブジェクトプログラムにより、I=1からI=100までの(L,I) (但し、L=1)となる単純項目処理モジュールAを呼び出す。図7において、1回目に呼び出される処理モジュールは、A(1,1)であり、2回目に呼び出される処理モジュールは、A(1,2)であり、100回目に呼び出される処理モジュールは、A(1,100)である。

【0006】つまり、図8に示す単純項目処理モジュールAがI=1~100まで100回、呼び出される。また、ユーザにより例2のようなFORTRANプログラムが作成された場合、

(例2)

6

```
REAL*4 A(10000),B(100)
L=1
DO I=1, 100
  B(I)=A(L,I)
ENDDO
WRITE(10) (B(I),I=1,100)
```

上記のようにユーザがプログラミングすれば、プログラムの最適化を行っており、オブジェクトプログラムは、図8に示すモジュールAを1回のみ呼び出すことで、例1と同じ結果を導出することができる。

【0007】また、以下に示すように、ライブラリの連続領域内に入出力対象項目をオブジェクトプログラムからアクセスする方法がある。図9は、従来の繰り返し処理を説明するための図である。FORTRANプログラムにおける入出力文を解析し、所定の入出力ライブラリ15を呼び出すオブジェクトプログラム12を出力するFORTRANコンパイラ11を備えたデータ処理システムにおいて、FORTRANコンパイラが、FORTRANプログラム内に入出力文において、繰り返し項目の並びを検出した時に、繰り返し回数と、1つの要素の長さ、及び位置情報を有する繰り返し項目パラメータ14を作成し、入出力ライブラリ15を呼び出すオブジェクトプログラム12を出力するように構成されている。このような構成において、入出力ライブラリ15は、実行時に、繰り返し項目パラメータ14を解析し、入出力対象項目が連続領域に並んでいることを条件として、配列処理化による複数項目に対する処理を行う。

【0008】これにより、プログラム内に入出力文に繰り返し項目の並びが現れた場合に、配列処理が可能な連続領域に入出力対象項目が並んでいる場合には、入出力ライブラリ15において高速な繰り返し処理を行うことが可能である。

【0009】

【発明が解決しようとする課題】しかしながら、上記従来の方式の例1のようなプログラムでは、入出力DO形反復の処理では、図8に示すようなモジュールの呼び出しを行うと、これらのモジュールはソースへの移植性とソフト部品の共通化を考慮して、処理モジュールがメインモジュールの外付けとなっているため、モジュール呼び出しのオーバーヘッドが大きくなり、プログラムの実行性能が劣化するという問題がある。

【0010】また、ユーザが上述の例2のようなプログラムによれば、実行結果が同じであるにも係わらず、モジュールを1回呼び出すことにより入出力DO項目の並びの処理を行っているため、実行時の性能がかなり異なる。しかし、このように、実行性能を向上させるには、ユーザのプログラミング技術に依ることになるため、プログラミングの質の均一性に問題がある。

【0011】このように、例1では、アクセスする変数が不連続領域であり、例2では、連続領域に対するアク

セスを行うことにより、同じ結果を取得する場合であってもアクセスのオーバーヘッドには大きな差異がある。しかし、このユーザに対しても例2のようなプログラミング技術を要求することは、ユーザの能力に依ることが大きいと、同一品質のプログラミング技術を期待することは、困難である。

【0012】また、従来の図9に示す例では、効率のよい繰り返し処理は、入出力対象項目が、連続領域に配置されていることが条件となっているため、不連続領域に配置されている入出力対象項目については、入出力文に指定されている回数分の繰り返し処理が必要となる。本発明は、上記の点に鑑みなされたもので、ユーザのプログラミング技術に依らず、プログラムで入出力並びに不連続領域に対する入出力D O形反復が記述された場合でも、当該不連続領域に格納されている値を再編することを可能とし、1回の処理モジュールの呼び出しで繰り返し処理を実行可能とし、システムを使用するアプリケーションソフトやエンドユーザに対して、より高速でかつ簡単なシステムを提供可能なプログラムの入出力反復処理方法及びシステム及びプログラムの入出力反復処理プログラムを格納した記憶媒体を提供することを目的とする。

【0013】更なる本発明の目的は、プログラムに記載されている不連続領域に対する入出力の反復処理である場合に、不連続領域に格納されている値の再編処理をユーザが選択可能なプログラムの入出力反復処理方法及びシステム及びプログラムの入出力反復処理プログラムを格納した記憶媒体を提供することである。

【0014】

【課題を解決するための手段】第1の発明は、FORTRAN言語、BASIC言語または、C言語を含むプログラム中に入出力反復の入出力が記載されており、入出力ライブラリ内の不連続領域をアクセスするプログラムの入出力反復処理方法において、プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定し、入出力対象項目要素が入出力ライブラリ内の不連続領域に存在する場合に、オブジェクトインターフェースに指定されたパラメータに基づいて、入出力ライブラリ内部で不連続領域の値を連続領域に置き換え、連続領域をアクセスするオブジェクトプログラムを制御する。

【0015】第2の発明は、オブジェクトインターフェース中のパラメータの1つの入力対象項目要素の長さと、増加アドレス値を比較し、1つの要素の長さと、増加アドレス値が一致しない場合には、入出力ライブラリ内の不連続領域の値を格納するための連続領域を確保し、確保された連続領域に、オブジェクトインターフェースに指定されたパラメータに基づいて不連続領域の値を複

写する。

【0016】第3の発明は、不連続領域の値を連続領域に格納する際に、指定された不連続領域の値の有効値のみを選択して、連続領域に格納する値とする。第4の発明は、不連続領域の値を連続領域に格納する際に、指定された連続領域へ置換する不連続領域の有効値から次の有効値までの配列要素数及び置換個数に基づいて、不連続領域から連続領域への置換を行う。

【0017】第5の発明は、FORTRAN言語、BASIC言語または、C言語を含むプログラム中に入出力反復の入出力が記載されており、入出力ライブラリ内の不連続領域をアクセスするプログラムの入出力反復処理方法において、プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定し、入出力対象項目が入出力ライブラリ内の不連続領域に存在する場合に、オブジェクトインターフェースに指定されたパラメータに基づいて、繰り返し対象項目をアクセスするモジュールに、入出力ライブラリ内部で有効となる項目のパラメータを渡す。

【0018】第6の発明は、繰り返し対象項目をアクセスするモジュールに、指定された不連続領域の値の有効値のみを選択して、連続領域に格納する値を渡す。第7の発明は、繰り返し対象項目をアクセスするモジュールに、指定された連続領域へ置換する不連続領域の有効値から次の有効値までの配列要素数及び置換個数を渡す。

【0019】第8の発明は、入出力ライブラリ内の不連続領域をアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定するパラメータ指定手段と、繰り返し項目が入出力ライブラリ内の不連続領域に存在する場合に、パラメータ指定手段により指定されたパラメータに基づいて、入出力ライブラリ内部で不連続領域の値を連続領域に置き換える連続領域置換手段と、連続領域をアクセスするようオブジェクトプログラムを制御するオブジェクトプログラム制御手段とを有する。

【0020】第9の発明において、連続領域置換手段は、オブジェクトインターフェース中のパラメータの1つの要素の長さと、増加アドレス値を比較する比較手段と、1つの要素の長さと、増加アドレス値が一致しない場合には、入出力ライブラリ内の不連続領域の値を格納す

るための連続領域を確保する連続領域確保手段と、確保された連続領域に、オブジェクトインターフェースに指定されたパラメータに基づいて不連続領域の値を複写する複写手段とを含む。

【0021】第10の発明において、連続領域置換手段は、不連続領域の値を連続領域に格納する際に、不連続領域の有効値を指定する有効値指定手段と、有効値指定手段により指定された不連続領域の値の有効値のみを選択して、連続領域に格納する値とする第1の置換値決定手段とを含む。第11の発明において、連続領域置換手段は、不連続領域の値を連続領域に格納する際に、連続領域へ置換する不連続領域の有効値から次の有効値までの配列要素数及び置換個数を指定する置換値指定手段と、置換値指定手段により指定された値に基づいて、不連続領域から連続領域への置換を行う値を決定する第2の置換値決定手段とを含む。

【0022】第12の発明は、入出力ライブラリ内の不連続領域をアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、プログラム中の入出力文において、繰り返し項目の並びを解析し、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定するパラメータ指定手段と、繰り返し項目が入出力ライブラリ内の不連続領域に存在する場合に、パラメータ指定手段により指定されたハラメータに基づいて、入出力ライブラリ内部で不連続領域のアクセス対象となる項目をアクセスするためのハラメータを生成するアクセス対象パラメータ生成手段と、アクセス対象パラメータ生成手段により生成されたハラメータを、繰り返し処理を行うオブジェクトプログラムに転送するパラメータ転送手段とを有する。

【0023】第13の発明は、アクセス対象パラメータ生成手段において、不連続領域内の繰り返し対象となる有効値を指定する有効値指定手段を含む。第14の発明は、アクセス対象パラメータ生成手段において、不連続領域内のある有効値から次の有効値までの配列要素数及び置換個数または、バイト数を指定する置換値指定手段を含む。

【0024】第15の発明は、入出力ライブラリ内の不連続領域をアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理シ

テムにおいて、プログラム中の入出力文において、繰り返し項目の並びを解析する解析プロセスと、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定させるパラメータ指定プロセスと、繰り返し項目が入出力ライブラリ内の不連続領域に存在する場合に、パラメータ指定プロセスにより指定されたパラメータに基づいて、入出力ライブラリ内部で不連続領域の値を連続領域に置き換える連続領域置換プロセスと、連続領域をアクセスするようオブジェクトプログラムを制御するオブジェクトプログラム制御プロセスとを有する。

【0025】第16の発明は、入出力ライブラリ内の不連続領域をアクセスするFORTRAN言語、BASIC言語または、C言語を含む入出力反復の入出力が記載されているプログラムと、該プログラムにおける入出力文を解析し、オブジェクトプログラムを出力するコンパイラと、該オブジェクトプログラムによりアクセスされるライブラリを有するプログラムの入出力反復処理システムにおいて、プログラム中の入出力文において、繰り返し項目の並びを解析する解析プロセスと、繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定させるパラメータ指定プロセスと、繰り返し項目が入出力ライブラリ内の不連続領域に存在する場合に、パラメータ指定プロセスにより指定されたパラメータに基づいて、入出力ライブラリ内部で不連続領域のアクセス対象となる項目をアクセスするためのハラメータを生成するアクセス対象パラメータ生成プロセスと、アクセス対象ハラメータ生成プロセスにより生成されたパラメータを繰り返し処理を行うオブジェクトに転送するパラメータ転送プロセスとを有することを特徴とするプログラムの入出力反復処理プログラムを格納した記憶媒体。

【0026】第1、第8及び第15の発明によれば、入出力反復を含む文、つまり、同じ処理をn回繰り返す記述を含むソースプログラムをコンパイルしてオブジェクトプログラムを出力する際に、当該オブジェクトプログラムか、入出力ライブラリ内のアクセス対象要素が不連続に格納されている不連続領域をアクセスするような場合に、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定して、当該ハラメータに基づいて、入出力ライブラリ内部で不連続領域の値を連続領域に置き換えることにより、当該連続領域をアクセスする下位プログラムを1回呼び出せばよいから、オーバーヘッドが大幅に削減することが可能となる。

【0027】第2及び第9の発明によれば、オブジェクトインターフェースに、少なくともライブラリ内の項目要素の1つの要素の長さと、増加アドレス値を比較し、

1つの要素の長さと、増加アドレス値が一致しない場合には、当該項目要素は、不連続であると判定し、入出力ライブラリの不連続領域の値を格納するための連続領域を確保し、確保された連続領域に、オブジェクトインターフェースに指定されたパラメータに基づいて不連続領域の値を複写することにより、上記の第1及び第8の発明が達成される。また、1つの要素の長さと増加アドレス値が一致する場合、即ち、ライブラリ内の項目要素が連続している場合には、従来の技術（特開平2-116930）を適用することにより、単純な処理の繰り返しを行う下位プログラムを1回呼び出すことにより対処できる。

【0028】第3及び第10の発明によれば、不連続領域の値を連続領域に格納する際に、ユーザにより指定された不連続領域の値のうち、連続領域に格納すべき値に基づいて有効値のみを選択して、連続領域に格納する値とすることにより、繰り返し処理の対象となる値を選択して格納することが可能となる。第4及び第11の発明によれば、不連続領域の値を連続領域に格納する際に、ユーザにより指定された連続領域へ置換する不連続領域内における有効値から次の有効値までの配列要素数及び置換個数に基づいて、不連続領域から連続領域への置換を行うことにより、規則的に連続領域に格納すべき要素を決定することが可能となる。

【0029】第5、第12及び第16の発明によれば、プログラム中に繰り返しが存在する場合には、繰り返し回数及び項目要素の長さ及び位置情報をパラメータとしてオブジェクトインターフェースに指定し、入出力対象項目が入出力ライブラリ内の不連続領域に存在する場合には、オブジェクトインターフェースに指定されたパラメータに基づいて、繰り返し対象項目をアクセスするモジュールに、入出力ライブラリ内部で有効となる項目要素のパラメータを渡すことにより、入出力ライブラリ内に連続領域を確保する必要がなく、アクセス対象となる項目要素を指し示すパラメータを繰り返し対象項目をアクセスするモジュールに直接渡すことが可能となる。

【0030】第6及び第13の発明は、繰り返し対象項目をアクセスするモジュールに、不連続領域の値のうち、繰り返し対象の有効値のみを選択して渡すことにより入出力ライブラリ内に連続領域が不要となる。第7及び第14の発明は、繰り返し対象項目をアクセスするモジュールに、指定された連続領域へ置換する不連続領域の有効値から次の有効値までの配列要素数及び置換個数を渡すことにより入出力ライブラリ内の連続領域が不要となる。

【0031】

【発明の実施の形態】図1は、本発明の反復処理システムの構成を示す。同図に示すシステムは、入力プログラムとなるソースプログラム30、ソースプログラム30をコンパイルしてオブジェクトプログラム100を生成

するコンパイラ40、オブジェクトインターフェースモジュール200、置換指示部300、ライブラリ400より構成される。

【0032】オブジェクトプログラム100は、置換制御部111を有する反復処理モジュール110と、当該反復処理モジュール110により呼び出される単純項目処理モジュール50より構成される。オブジェクトインターフェースモジュール200は、コンパイラ40から渡された入力プログラム30の反復処理の解析結果である反復処理対象の1要素の長さと、増加アドレス値を保持し、反復処理モジュール110に渡す。

【0033】ライブラリ400は、配列要素が不連続に格納されている不連続領域410と、置換制御部111により不連続領域410の要素を抽出して格納するための連続領域420を有する。置換指示部300は、ユーザにより不連続領域410から連続領域420に置換する際の指示を行う。

【0034】上記の構成に基づいて、反復処理の動作を説明する。図2は、本発明の動作を説明するためのフローチャートである。

ステップ101）オブジェクトプログラム100の反復処理モジュール110は、オブジェクトインターフェース200中の1要素の長さと、増加アドレス値を比較する。

【0035】ステップ102）1要素の長さと増加アドレス値が同じ値である場合には、ライブラリ400内の配列要素が連続していることを意味するため、図9に示す従来の方法と同様のプログラムの最適化処理を行う。また、異なる値である場合には、配列要素が不連続であることを意味するため、ステップ103に移行する。

【0036】ステップ103）置換指示部300から実行時のオプション指定があるか否かを判定する。

ステップ104）置換指示部300からの指示がある場合には、ステップ105に移行し、ない場合には、図9に示す方法により、連続領域をアクセスする従来の処理を行う。

【0037】ステップ105）置換制御部111は、ライブラリ400内に、不連続領域410の要素を格納するための連続領域420を確保する。

ステップ106）不連続領域410の配列の要素が0以下である場合にはステップ109に移行し、1以上の場合にはステップ107に移行する。

ステップ107）置換制御部111は、不連続領域410の配列の1つの要素を連続領域420に転送し、格納する。

【0038】なお、置換指示部300から有効値、増加アドレス、増加バイト数等の指定がある場合には、このような指定条件に基づいて、不連続領域410の配列要素を連続領域に格納する。



ステップ108) オブジェクトインターフェースモジュール200より増加アドレス値を取得して、要素数に加算し、ステップ106に移行する。

【0039】ステップ109) 単純項目処理モジュール50を呼び出し、単純項目処理モジュール50は、ライブラリ400の連続領域420にアクセスして処理を実行する。これにより、入出力対象項目がライブラリ400に連続的に配置されている場合には、不連続領域410から連続領域420に配列要素を移管する必要はなく、前述の従来の図9に示す方法を用いて実行し、不連続領域410に配置されている場合には、入出力対象項目を不連続領域410から連続領域420に置換することにより、図9に示すように、連続領域に対する処理を行えばよい。

【0040】

【実施例】以下、本発明の実施例を図面と共に説明する。以下の実施例において、プログラム言語をFORTRANとして説明する。

【第1の実施例】第1の実施例は、反復処理モジュール110に、ユーザから、不連続領域410から連続領域420に処理対象となる項目を置換する指示が置換指定部150から投入された場合の例を示す。

【0041】図3は、本発明の第1の実施例のFORTRANプログラムの繰り返し処理を説明するための図である。FORTRANプログラムの例として、

```
J=1L
WRITE(*,*) (I(J,K)K=1,10)
WRITE(10) (A(L,I),I=1,100)
↓
WRITE(*,*) (X(K)K=1,10)
```

が提供されている場合に、オブジェクトインターフェースモジュール200は、コンパイラ40から渡されたパラメータとして

1要素の長さ=4

配列の先頭番地=1(1,1)

増加アドレス値=400

を保持しているものとする。このとき、反復処理モジュール110の置換制御部111は、オブジェクトインターフェースモジュール200内の1要素の長さ、増加アドレス値が等しいかを判定し、等しくない時には、不連続領域を格納する領域を格納する。なお、ユーザにより、不連続領域410から連続領域420に置換する指示が指定され、かつ、種々のオプション指示が入力されたものとする。なお、オプション指示に対する動作は後述する。

【0042】図3の例では、不連続領域410から連続領域420に複写される値は、上記のパラメータに基づいて、

1(1,1), 1(1,2), 1(1,3), 1(1,4), 1(1,5), 1(1,6), 1(1,7), 1

(1,8), 1(1,9), 1(1,10)

となり、これらの値を、連続領域420の

X(1), X(2), X(3), X(4), X(4), X(5), X(6), X(7), X(8), X(9), X(10)

に複写する。

【0043】これにより、反復処理モジュール110

は、単純項目処理モジュール50に対して、実行指示を発行する。単純項目処理モジュール50は、ライブラリ400の連続領域420の入出力対象項目にアクセスする。つまり、反復処理モジュール110は、連続領域420を生成し、単純項目処理モジュール50を1回呼び出せばよい。

【0044】【第2の実施例】次に、第2の実施例として、置換指示部300から、置換実行指示と共に、不連続領域410に格納されている無効値を除去し、有効値のみをアクセス可能なように、連続領域420に置換するような指示が発行されたものとする。本実施例では、不連続領域410に格納されている処理対象項目を選択して、連続領域420に格納するのではなく、不連続領域410に格納されている全ての処理対象項目を連続領域420に転送し、当該連続領域420に格納する際に、個々の項目が有効値であるか無効値であるかを判定して、いずれかの表示を付与するものである。

【0045】図4に示すように、連続領域420に変換すべき値を有効値(VARID DATA)としたとき、配列の全要素数から無効値(INVARID DATA)を求め、有効値の数と無効値の数を比較し、その大小により、どちらかの値表示を行い、単純項目処理モジュール50に渡す。同図の例では、配列要素数100個に対して、10個の有効値、90個の無効値が求められ、有効値と無効値の数を比較し、その大小により、どちらかの値表示(INVARID/VARID)を行い、単純項目処理モジュール50に制御を渡す。本実施例では、数が少ない方の表示を用いるものとする。従って、“VARID DATA”を付与するものとする。

【0046】同図の例は、不連続領域410の全ての項目を連続領域420に置換し、そのうち、

1(1,1), 1(1,2), 1(1,3), 1(1,4), 1(1,5), 1(1,6), 1(1,7), 1(1,8), 1(1,9), 1(1,10)

に“VARID DATA”を付与する。

【0047】これにより、単純項目処理モジュール50は、“VARID DATA”が付与されている項目のみにアクセスすることかてきる。

【第3の実施例】次に、本実施例では、不連続領域410から連続領域420への置換は行わずに、反復処理モジュール110は、ライブラリ400に格納されている配列において、変換すべき値を有効値としたとき、有効値から次の有効値までの配列要素数及び置換個数を単純項目処理モジュール50に渡す。これらのパラメータ

は、ユーザにより置換指示部300から指示されるものとする。

【0048】図5は、本発明の第3の実施例の単純項目処理モジュールにパラメータを渡す例（その1）を示す図である。同図に示す配列はライブラリ400の例であり、同図において、変換すべき値を有効値としたとき、有効値から次の有効値までの配列要素数及び置換個数を単純項目処理モジュール50に渡す。項目1（1，1）から1（1，2）の場合、この間の配列要素数は9個、変換個数は10個である。

【0049】つまり、反復処理モジュール110は、単純項目処理モジュール50に対して、先頭の有効値と次の有効値の間に9個の無効値があり、全体の変換個数は、10個である旨のパラメータを渡す。これにより、単純項目処理モジュール50は、当該パラメータに基づいて、

1（1，1）、1（1，2）、1（1，3）、1（1，4）、1（1，5）、1（1，6）、1（1，7）、1（1，8）、1（1，9）、1（1，10）

にアクセスする。

【0050】なお、単純項目処理モジュール50では、ワークメモリ（図示せず）内に、パラメータと共に、取得した有効値を保持するようにしてもよい。

【第4の実施例】本実施例では、反復処理モジュール110において、変換すべき値を有効値としたとき、有効値から次の有効値までの増加アドレス及び置換最終アドレスを単純項目処理モジュール50に渡す処理を行う。

【0051】図6は、本発明の第4の実施例の単純項目処理モジュールにパラメータを渡す例（その2）を示す図である。同図において、反復処理モジュールは、1（1，1）から1（1，2）の場合、この間の増加アドレス値は、 $9 \times 4 = 36$ バイトであり、置換最終アドレス1（9，1）を単純項目処理モジュール50に渡す。

【0052】これにより、単純項目処理モジュール50は、渡されたパラメータに基づいて、ライブラリ400の項目を取得する。なお、本実施例においても第3の実施例と同様に、単純項目処理モジュール50において、ワークメモリ（図示せず）内に取得したパラメータおよび処理対象項目を保持するようにしてもよい。

【0053】このように、上記の第1と第2の実施例によれば、処理対象の項目が不連続領域に格納されて、当該不連続領域をアクセスする場合に、ライブラリ400内部で連続領域に置き換えて、反復処理モジュール110から当該モジュール110の下位モジュールである単純項目処理モジュール50を1回呼び出せば済むことになる。

【0054】また、第3と第4の実施例によれば、ライブラリ400内部で有効値となる値の不連続領域から連続領域に対する置換を行わずに、単純項目処理モジュール50に対して、有効値を抽出するためのパラメータを渡

すことにより、前述の第1及び第2の実施例と同様の効果を得ることが可能となると共に、ライブラリ400内に連続領域420が不要となる。

【0055】このように、本発明によれば、不連続領域から連続領域に移管してアクセスを行う場合、不連続領域に格納されている要素を、指定されたパラメータに基づいて単純項目処理モジュール50に直接通知するような方法にいずれにおいても、オブジェクトから1回だけ単純項目処理モジュール50を呼び出せばよいだけである。

【0056】従って、ユーザが最適化されたプログラミング技術を習得していなくとも、このような処理を行うことにより、ユーザにプログラミング時における負担をかけずに、オブジェクトから繰り返し処理のためのモジュールを1回だけ呼び出すことができる。なお、上記の実施例では、反復処理システムとして説明したが、この例に限定されることなく、オブジェクトプログラム100、オブジェクトインタフェースモジュール200、置換指示部300の制御プログラムをフロッピーディスクやCD-ROM等の可搬記憶媒体に格納し、反復処理システムを必要とするコンピュータにインストールすることが可能であると共に、ディスク装置等に記憶させ、反復処理プログラムを読み出すことにより利用することも可能である。

【0057】なお、上記の実施例では、FORTRANプログラムにおける反復処理を例として説明したが、この例に限定されることなく、BASIC言語、C言語等においても同様の処理を適用することが可能である。なお、本発明は、上記の実施例に限定されることなく、特許請求の範囲内で種々変更・応用が可能である。

【0058】

【発明の効果】上述のように、本発明のプログラムの反復処理方法及びシステムによれば、プログラムで入出力及び入出力反復を意味する文が記述されている場合、アクセスするライブラリ内の領域が不連続領域であっても、ライブラリ内部で連続領域に置き換えて、入出力反復処理における処理モジュールの呼出し回数を1回とすることが可能である。

【0059】これにより、FORTRANプログラム、BASICプログラム、または、Cプログラム等のプログラムの実行時のCPU性能が高速化し、かつ、ユーザに対してプログラムの同一化を保証することができる。即ち、より高速なシステムを提供し、プログラムの複雑化を防ぐことができる。つまり、FORTRAN等のシステムを使用するアプリケーションソフトやエンドユーザに対し、より高速でかつ簡単なシステムを提供し、ユーザがより品質の高いプログラムを作成するための機能を向上させることが可能となり、処理プログラムとして要求される高品質化、高性能化を図ることができる。

【図面の簡単な説明】

【図1】本発明の反復処理システムの構成図である。

【図2】本発明の反復処理を説明するためのフローチャートである。

【図3】本発明の第1の実施例のFORTRANプログラムの繰り返し処理を説明するための図である。

【図4】本発明の第2の実施例のFORTRANプログラムの繰り返し処理を説明するための図である。

【図5】本発明の第3の実施例の単純項目処理モジュールにパラメータを渡す例(その1)である。

【図6】本発明の第4の実施例の単純項目処理モジュールにパラメータを渡す例(その2)である。

【図7】従来の反復処理を説明するための図である。

【図8】オブジェクトプログラムの呼出し方を説明するための図である。

\*

\*【図9】従来の繰り返し処理を説明するための図である。

【符号の説明】

30 入力プログラム

40 コンパイラ

50 単純項目処理モジュール

100 オブジェクトプログラム

110 反復処理モジュール

111 置換制御部

200 オブジェクトインターフェースモジュール

300 置換指示部

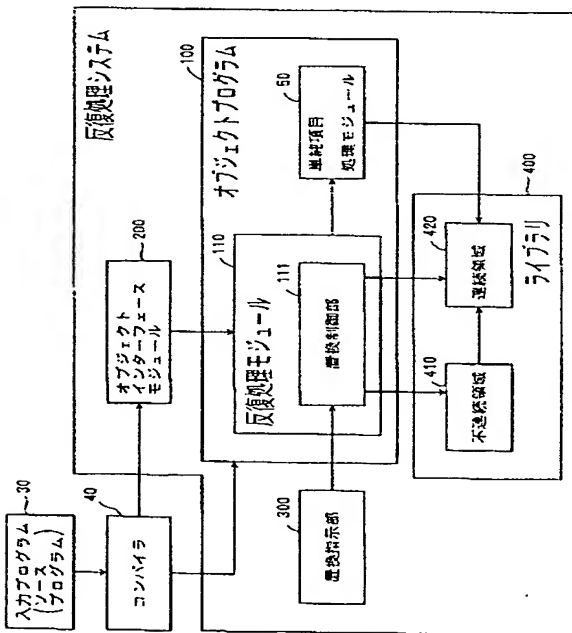
400 ライブラリ

410 不連続領域

420 連続領域

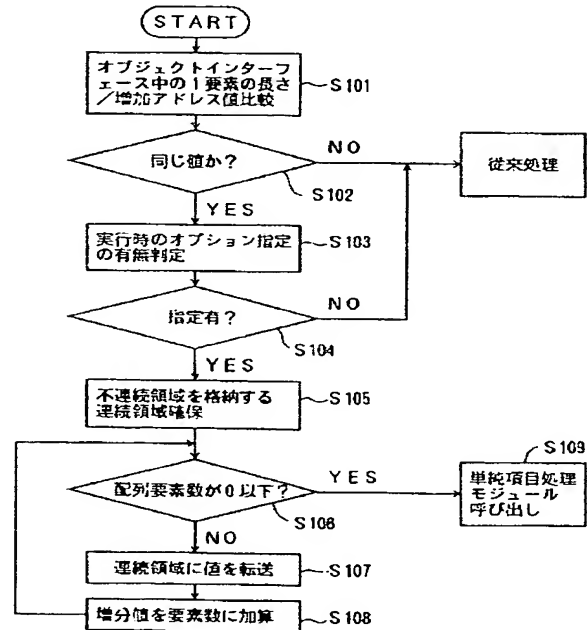
【図1】

本発明の反復処理システムの構成図



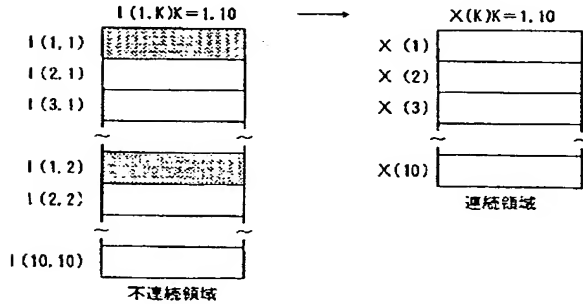
【図2】

本発明の反復処理を説明するためのフローチャート



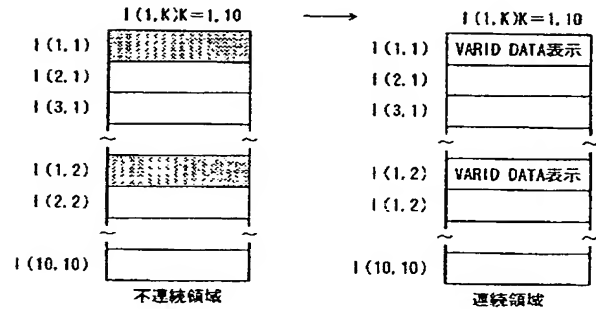
【図 3】

本発明の第1の実施例のFORTRANプログラムの  
繰り返し処理を説明するための図



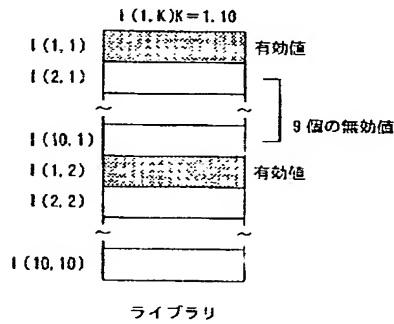
【図 4】

本発明の第2の実施例のFORTRANプログラムの  
繰り返し処理を説明するための図



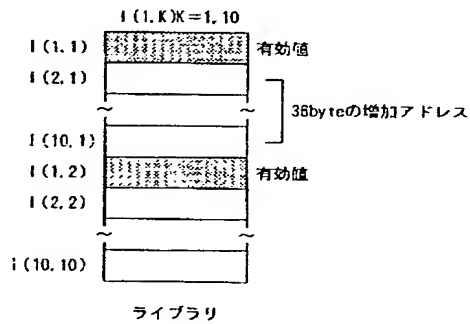
【図 5】

本発明の第3の実施例の単項項目処理モジュールに  
パラメータを渡す例（その1）



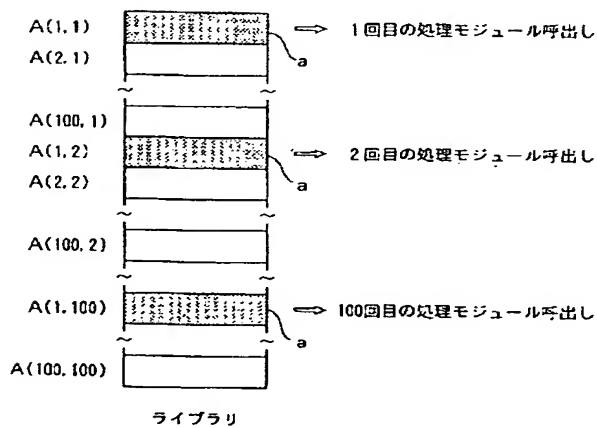
【図 6】

本発明の第4の実施例の単項項目処理モジュールに  
パラメータを渡す例（その2）



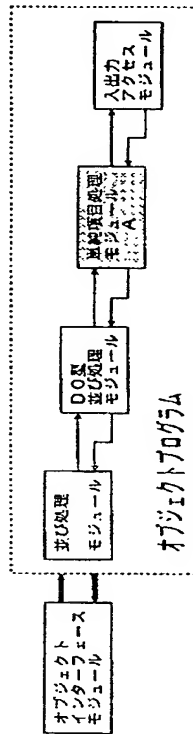
【図 7】

従来の反復処理を説明するための図



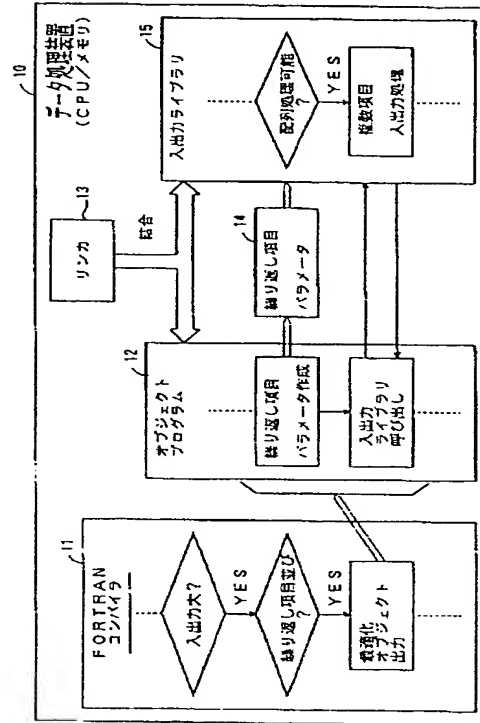
【圖8】

オブジェクトプログラムの呼び出し方を説明するための図



【図9】

従来の繰り返し処理を説明するための図



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**